शोध आलेख / Research Article

नैचुरल लैंग्वेज प्रोसेसिंग में ऊर्दू स्टेमर का विकास

Development of Urdu Stemmer in Natural Language Processing

वैशाली गुप्ता[1], निशीथ जोशी[2], इति माथुर[3]

[1] आई.पी.एस. एकैडमी, आई.ई.एस., इंदौर, मध्य प्रदेश

[2,3] आपाजी संस्थान, वनस्थली विद्यापीठ, राजस्थान

[1]vaishali.gupta77@gmail.com, [2] nisheeth.joshi@rediffmail.com, [3] mathur_iti@rediffmail.com

सारांश

स्टेमिंग (Stemming) की प्रक्रिया मार्फोलोजिकल एनालाईजर (Morphological Analyser), इन्फोर्मेशन रिट्रीवल (Information Retrieval), नैचुरल लैंगुएज प्रोसेसिंग (Natural Language Processing)एवं स्पेल चेकिंग (Spell Checking) इत्यादि में बहुत महत्वपूर्ण स्थान रखती है। स्टेमिंग के द्वारा हम किसी भी शब्द के "'स्टेम" (आधार अथवा मूल) को प्रथक कर सकते हैं। दूसरे शब्दों में हम कह सकते हैं कि प्रेफिक्स (prefix) (उपसर्ग) एवं /अथवा सफिक्स (suffix) (प्रत्यय) को किसी शब्द से हटा कर 'मूल' शब्द प्राप्त करना ही स्टेमिंग है। स्टेमिंग से प्राप्त मूल शब्द इन्फ्लेक्श्नल (Inflectional) अथवा डेरिवेशनल मार्फोलोजी (Derivational Morphology) को प्रदर्शित करता है। मॉर्फोलोजी से अर्थ है किसी भी शब्द की सम्पूर्ण रचना। इन्फ्लेक्श्नल मार्फोलोजी में मूल शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी के समान होती है। डेरिवेशनल मार्फोलोजी में मूल शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी से भिन्न होती है। इस शोध पत्र में, हमने नियमबद्ध इन्फ्लेक्श्नल एवं डेरिवेशनल ऊर्दू स्टेमर के विकास पर दृष्टि डाली है। यहाँ हमने लोंगेस्ट सफिक्स स्ट्रिपिंग एल्गोरिथम (Longest Suffix Stripping)का इस्तेमाल किया है जिसके द्वारा किसी भी शब्द एवं वाक्यों में से 'स्टेम' शब्द का प्रथकीकरण किया जा सकता है।

Abstract:

The process of 'Stemming' plays an important role in the field of Natural Language Processing, Morphological Analyzer, Spell Checker and Information Retrieval. Through the Stemming, we can extract 'root' or 'stem' part from the given word. In other words, we can say that Stemming is a process of removing suffix and prefix from 'root' or 'stem' word. These obtained 'root' words display inflectional or derivational morphology. Morphology means the complete structure of any word. In Inflectional morphology, category of 'root' word is similar to its actual word and category of 'root' word is different to its actual or given word in derivational morphology. In this research paper, we focus on the development of inflectional and derivational rule based Urdu Stemmer. Here we are using Suffix Stripping algorithm for the extraction of 'Stem' or 'Root' word from given word.)

सूचक–शब्दः स्टेम, इन्फ्लेक्शनल, डेरिवेशनल, मार्फोलोजी, अफिक्स, ऊर्दू।

Keywords: stem, inflectional, derivational, morphology, affix, Urdu

1. परिचय

आज के युग में कम्प्यूटर का महत्वपूर्ण योगदान है। हर तरह के कार्य कम्प्यूटर के द्वारा आसानी से किये जा सकते हैं। अब हम यदि भाषा की बात करें तो हम पार–बहुभाषी (cross-lingual) विश्व में रह रहे

हैं जहाँ हर व्यक्ति को हर भाषा का ज्ञान हो यह आवश्यक नहीं है। इस भाषा रुपी समस्या के समाधान के लिए शोधकर्ताओं ने "मशीन अनुवाद" नामक तकनीक को प्रस्तावित किया। इस मशीन अनुवादक सॉफ्टवेयर के द्वारा एक भाषा को दूसरी भाषा में परिवर्तित कर सकते हैं, जिससे हम कहीं भी किसी भी देश के बहुभाषाबद्ध (multilingual) लोगों से जुड़े रह सकते हैं।

इस मशीन अनुवादक के विकास में बहुत सारी समस्याएं उभर कर आईं। सर्वप्रथम समस्या शब्दों के रूपात्मक विश्लेषण की आई। रूपात्मक विश्लेषण के द्वारा किसी भी शब्द की व्याकरण संबंधी जानकारी प्राप्त की जा सकती है। इसकी दो विधियाँ हैं : इन्फ्लेक्शनल एवं डेरिवेशनल। इन्फ्लेक्शनल रूप में स्टेम शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी के समान होती है एवं डेरिवेशनल रूप में स्टेम शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी से भिन्न होती है। इन दोनों विधियों की प्रक्रिया को पूर्ण करने के लिए स्टेमर का विकास किया गया है।

स्टेमिंग एक तरह की प्रक्रिया है जिसमें अफिक्स (प्रत्यय) अपने स्टेम शब्द से अलग हो जाते हैं। अफिक्स के अंतर्गत ही प्रेफिक्स एवम् सफिक्स आते हैं। प्रेफिक्स वो होते हैं जो शब्द की शुरुआत में जुड़े होते हैं और सफिक्स शब्द के अंत में जुड़ा भाग होता है। जबकि स्टेम अपने आप में शब्द का आधार है। उदाहरण के लिए– प्रसन्नता, प्रसन्नमयी, प्रसन्नचित एवं अप्रसन्न, इन शब्दों के अफिक्स 'ता', 'मयी', 'चित' एवं 'अ' होंगे। इन सभी शब्दों का आधार यानि स्टेम 'प्रसन्न' होगा। यह उभयनिष्ठ आधार, विभिन्न मार्फोलोजिकल शब्दों से सम्बंधित है। इसी प्रकार ऊर्दू भाषा में – بےصبر (*बेसब्र*), بےصری (*बेसब्री*) एवं سبردار (*सब्रदार*) शब्दों के अफिक्स 'بے (*बे*)' ی (*ई*)'एवं 'دار (*दार*)' होंगे और इनका स्टेम 'صبر (*सब्र*)' होगा।

स्टेमर वास्तव में एक एल्गोरिथ्म (कलन विधि) है, जिसके माध्यम से हम शब्दों के स्टेम को अलग कर सकते हैं। यह नैचुरल लैंगुएज प्रोसेसिंग में महत्वपूर्ण स्थान रखती है। यह पद्धति स्पेल चेकिंग, मशीन अनुवाद के मूल्यांकन और इन्फोर्मेशन रिट्रीवल में विशेषत: इस्तेमाल की जाती है। यहाँ हम ऊर्दू भाषा के लिए एक नियमबद्ध स्टेमर प्रस्तुत करने जा रहे हैं। ऊर्दू भाषा में कई तरह की चुनौतियों का सामना करना पड़ता है। ऊर्दू दुर्बल इन्फ्लेक्शनल भाषा है और इसकी जटिल एवं समृद्ध मार्फोलोजी एवं इसकी विविधतापूर्ण प्रकृति के कारण, स्टेमर का ऊर्दू में विकास चुनौतीपूर्ण है। यहां पर इंफ्लेक्शन को हम शब्द निर्माण की प्रक्रिया के तौर पर समझ सकते हैं, जिसके अंतर्गत हम एक शब्द को विभिन्न व्याकरण की श्रेणियों में संशोधित कर सकते हैं किंतु ऊर्दू भाषा को हम दुर्बल इंफ्लेक्शन इसलिये कहते हैं क्यूँकि ऊर्दू में प्राय् शब्द के इन्फ्लेक्शन मौजूद नहीं होते हैं। उदाहरण के तौर पर – मशहूर : एकवचन एवं मशाहीर : बहुवचन। इसकी जटिल मार्फोलोजी के कारण, इन शब्दों में कोई भी इन्फ्लेक्शन मौजूद नहीं है। अत: इन शब्दों में से स्टेम शब्द को परित्यक्य करना संभव नहीं है। इस प्रकार के शब्द अपवाद में गिने जाते हैं। किन्तु कुछ उदाहरण जैसे– बदमिजाज, बदमिजाजी एवं मिजाज–आसना शब्द को हम इन्फ्लेक्शनल या डेरिवेशनल शब्दों में गिन सकते हैं। चूँकि इन शब्दों में से हम इनका स्टेम अलग कर सकते हैं। साथ ही यहाँ पर समृद्ध मार्फोलोजी से तात्पर्य है कि ऊर्दू भाषा में विभिन्न व्याकरण की श्रेणियां हैं जैसे कि संज्ञा (ऊर्दू में 'इस्म'), सर्वनाम ('जमीर') इत्यादि।

शब्दों से उनके स्टेम भाग को पृथक करने के अंतर्गत अंडर स्टेमिंग (Over Stemming) एवं ओवर स्टेमिंग (Over Stemming) की समस्याएं आती हैं। यदि हम किसी अफिक्स को उसके मूल से विरक्त कर दें जबकि उस शब्द से वास्तव में किसी भी अफिक्स को विरक्त करने की आवश्यकता ही ना रही हो तो ऐसी समस्या को अंडर स्टेमिंग में शामिल करते हैं। उदाहरण के लिए - 'پیشگی' (*पेशगी*) शब्द से यदि अफिक्स 'ی (*ई*)' हटा देते हैं तो स्टेम शब्द 'پیشگ' (*पेशग*) प्राप्त होता है जो की अर्थहीन है। इस

'**پیشگ**' (*पेशग*) की बजाय हमें स्टेम '**پیش**' (*पेश*) एवं अफिक्स '**گی**' (*गी*) प्राप्त होना चाहिए था। अर्थात बहुत ही छोटे अथवा कम शब्द को किसी पूर्ण शब्द से हटाना, अंडर स्टेमिंग के अंतर्गत आएगा। इसके विपरीत ओवर स्टेमिंग में जिन अफिक्स को नहीं हटाना चाहिए वो भी हट जाते है। उदाहरण के तौर पर '**بدماش**' (*बदमाश*) शब्द से अगर '**بد**' (*बद*) अलग हो जाये एवं स्टेम शब्द '**ماش**'(माश) बचता है तो इस स्टेम शब्द '**ماش**'(माश) का कोई तात्पर्य नहीं है। यद्यपि इस शब्द से हमें प्रेफिक्स हटाने की आवश्यकता ही नहीं थी। ओवर स्टेमिंग में प्राय: अधिक से अधिक लम्बाई के अफिक्स के हट जाने की समस्या उभर कर आती है।

2. साहित्य की समीक्षा

नैचुरल लैंगुएज प्रोसेसिंग के क्षेत्र में, प्रथम स्टेमर लोविंस (Lovins) [1] के द्वारा 1968 में बनाया गया था। उन्होंने अंग्रेजी शब्दों से उनका स्टेम शब्द पृथक करने के लिए 260 नियमों को प्रस्तावित किया था। पोर्टर (Porter) [2] ने विभिन्न अफिक्सो के आधार पर स्टेमर को विकसित किया था। इस स्टेमर की प्रक्रिया पाँच चरण में पूर्ण होती थी। पहले चरण में, इन्फ्लेक्श्नल अफिक्स को नियंत्रित करते थे। अगले तीन चरण में, डेरिवेशनल सफिक्स को नियंत्रित किया एवं अंतत: आखिरी चरण में रिकॉडिंग होती थी। यहाँ पर भी इन्फ्लेक्श्नल मार्फोलोजी से तात्पर्य है कि स्टेम शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी के समान होती है एवं डेरिवेशनल मार्फोलोजी में स्टेम शब्द की श्रेणी उसके वास्तविक शब्द की श्रेणी से भिन्न होती है। खोजा एवं गर्सिदे (Khoja and Garside) [3] ने अरैबिक स्टेमर बनाया था जिसे सुपिरिओर रूट आधारित स्टेमर कहते थे। इस एल्गोरिथम में प्रेफिक्स, सफिक्स एवं इन्फिक्स तीनों उनके अफिक्स लिस्ट से मैप करा के पृथक कर लेते थे एवं स्टेम शब्द को अलग प्रदर्शित कर देते थे। विशेषत: संज्ञा के साथ इसे विभिन्न समस्याओं का सामना करना पड़ता था। 20 वीं सदी के पश्चात, शोधकर्ता मजूमदार एवं उनके साथियों (Majumder) [5] ने स्ट्रिंग्स के बीच की दूरी के आधार पर एक दृष्टिकोण पेश किया जो कि समूहों पर आधारित था और साथ ही इसे किसी भाषाई ज्ञान की आवश्यकता नहीं थी। इस दृष्टिकोण या प्रणाली को किसी भी भाषा के साथ लागू किया जा सकता है। अलश्मारी एवं लिन (Al-Shammari and Lin) [5] ने एजुकेटेड टेक्स्ट स्टेमर का प्रस्तुतीकरण किया। यह बहुत ही सरल, शब्दकोश मुक्त एवं कुशल स्टेमर है जिसके द्वारा स्टेमिंग की त्रुटियों में गिरावट आई और साथ ही इसे कम स्टोरेज एवं कम प्रोसेसिंग टाइम की जरुरत पड़ती है। अकरम (Akram)[5] एवं उनके साथियों ने ऊर्दू भाषा के लिए एक स्टेमर प्रस्तुत किया। यह स्टेमर सिर्फ ऊर्दू भाषा के लिए प्रतिबंधित है यानि इस स्टेमर के द्वारा हम दूसरी किसी भाषा जैसे हिंदी, अंग्रेजी, फारसी के शब्दों से उनके स्टेम को पृथक नहीं कर सकते हैं। इस स्टेमर के द्वारा हम प्रेफिक्स एवं सफिक्स दोनों को स्टेम शब्द से पृथक करा सकते हैं। खान एवं उनके साथियों (Khan et. al.)[7] ने नियम– आधारित स्टेमर बनाने में आने वाली कई चुनौतियों को पेश किया है। इन लोगों ने अरबी, फारसी और ऊर्दू भाषा के लिए नियमों पर आधारित बने हुए स्टेमर की भी चर्चा की है। इस नियमों पर आधारित स्टेमर के द्वारा प्रेफिक्स एवं सफिक्स अलग किये गए हैं और इन्हें नियंत्रित करने के लिए कई पैटर्न निश्चित किये गये हैं। प्रेफिक्स वो होते हैं जो शब्द की शुरुआत में जुड़े होते हैं और सफिक्स शब्द के अंत में जुड़ा भाग होता है। मिश्रा एवं प्रकाश (Mishra and Prakash)[8] ने हिंदी भाषा के लिए एक प्रभावी स्टेमर का प्रस्ताव रखा। यह स्टेमर ओवर स्टेमिंग एवं अंडर स्टेमिंग की समस्या का निवारण करता है और इसके द्वारा 91.59 % की सटीकता प्राप्त होती है। हुसैन (Husain) [9] ने ऊर्दू और मराठी भाषा के स्टेमर के विकास के लिए एक अनसुपरवाईजड द्रष्टिकोण का प्रस्ताव रखा। इस द्रष्टिकोण में, उन्होंने सफिक्स प्रथकीकरण के दो तरीकों का प्रस्ताव किया : 1) फ्रीक्वेंसी /आवृत्ति

आधारित सफिक्स स्ट्रिपिंग एल्गोरिथम (Frequency based Suffix Stripping Algorithm) और 2) लम्बाई आधारित सफिक्स स्ट्रिपिंग एल्गोरिथम (Length based Suffix Stripping Algorithm)। जूही एवम् उनके साथियों (Juhi)[10] गुजराती – हिंदी मशीन अनुवाद की गुणवत्ता को बढ़ाने के लिये पोस (Part-of-Speech) टैगिंग एवं स्टेमिंग का इस्तेमाल किया है। इन्होंने 5400 पोस टैग आधारित वाकयों पर 202 अफिक्स नियमों को लागू कर के अनुवाद किया तो इस प्रणाली की दक्षता 93.09 % मापी गई। स्निग्धा एवम् उनके साथियों (Snigdha)[11] ने नियमबद्ध हिंदी लेमेटाइजर को विकसित किया है। लेमेटाइजर के द्वारा हम स्टेमिंग में आने वाली कमियों को भी दूर कर सकते हैं। इसमें सफिक्स हटाने के साथ साथ कुछ नियम जोड़ने के लिये भी बनाये गये हैं। इस हिन्दी लेमेटाइजर की शुद्धता 89.08% मापी गई। मुबाशिर एवम् उनके साथियों (Mubashir) [12] ने रुपात्मक ऊर्दू की चुनौतियों को दूर करने के लिये एक प्रभावी प्रस्ताव दिया जिसमें नियम आधारित ऊर्दू स्टेमर की संरचना को दर्शाया गया है। यह स्टेमर ऊर्दू के स्टेम शब्द को उत्पन्न करता है और साथ ही कुछ दूसरी भाषाओं के स्टेम शब्दों को भी उत्पन्न करता है। जब्बर एवम उनके साथियों (Jabbar) [13] ने ऊर्दू भाषा के लिये कई संसाधनों का विश्लेषण और विकास किया। साथ ही विभिन्न प्रकार की स्टेमिंग के तरीकों को भी परिभाषित किया हुआ है। 2019 में फिर से मुबाशिर एवम् उनके साथियों (Mubashir) [14] ने एक अन्य व्यापक स्टेमर का विकास किया जो कि खास तौर पर रुपात्मक तरीके से समृद्ध ऊर्दू भाषा के लिये बनाया गया है। यह स्टेमर नियमबद्ध अफिक्स हटाने की पद्धति पर आधारित है जिसके द्वारा स्टेम शब्द को प्राप्त किया जा सकता है।

3. ऊर्दू की भाषाई पृष्ठभूमि/ऊर्दू की मार्फोलोजी

ऊर्दू भाषा अन्य इन्डो–आर्यन भाषाओं के समान ही है। इसे दायीं से बायीं तरफ फारसी–अरबी लिपि के समान लिखा जाता है। ऊर्दू एक मुक्त शब्द क्रम (free word order) एवं कमजोर विभक्ति वाली भाषा है। इसकी मार्फोलोजी बहुत ही जटिल एवं सम्रद्ध है। मार्फोलोजी का मतलब शब्दों की पूर्णत: बनावट अथवा आकृति विज्ञान से है। इसके द्वारा शब्दों को संज्ञा, क्रिया 'विशेषण इत्यादि शाब्दिक वर्गों में भी विभाजित किया जा सकता है। इसकी सबसे छोटी यूनिट 'मोर्फीम' अथवा 'लीमा' कहलाती है। मोर्फीम किसी भाषा की अर्थपूर्ण इकाई है जिसे आगे विभाजित नहीं किया जा सकता है। (Morpheme =fie Phoneme= Lofue) उदहारण के लिए शब्द –'آدم (आदम)' एक एकल मोर्फीम है किन्तु 'آدم زاد (आदमजाद)' शब्द में 'آدم (आदम)' एवं 'زاد (जाद)' दो मोर्फीम है। इस उदाहरण से हम समझ सकते हैं कि कुछ शब्द उसके दो वर्ग : स्टेम तथा अफिक्स से मिलकर बने होते हैं। यहाँ 'स्टेम' ही किसी शब्द का मुख्य मोर्फीम होता है। चूँकि अफिक्स के बजाय, स्टेम से ही हमें शब्द के अर्थ का ज्ञान हो पता है। यहाँ शब्दों को मोर्फीम कि सहायता से बनाने के दो व्यापक तरीके हो सकते हैं : इन्फ्लेक्श्नल मोर्फीम एवं डेरिवेशनल मोर्फीम। इन्फ्लेक्श्नल मोर्फीम में, वास्तविक शब्द एवं उसके स्टेम शब्द की श्रेणी समान होती है जैसे कि 'سوال +ات = سوالات (सवाल़आत=सवालात)' जहाँ 'سوال (सवाल)' एकवचन संज्ञा एवं 'سوالات (सवालात)' बहुवचन संज्ञा है। अर्थात इनकी श्रेणी समान ही है। डेरिवेशनल मार्फोलोजी में, वास्तविक शब्द एवं प्राप्त स्टेम शब्द या किसी शब्द में अफिक्स जोड़ने से प्राप्त नए शब्द की श्रेणी एक दूसरे से भिन्न होती है। उदहारण के तौर पर – 'حکمت+ی=حکمتی (हिकमत+ई=हिकमती)', यहाँ शब्द की श्रेणी संज्ञा (**اسم**) से विशेषण (**صفت**) में परिवर्तित हो गई है।

ऊर्दू मोर्फोलोजी में, संज्ञा, क्रिया, विशेषण एवं क्रिया–विशेषण के अनेक इन्फ्लेक्शन रूप हो सकते हैं। इसके अंतर्गत लिंग, एकवचन, बहुवचन, काल इत्यादि भी शामिल है। हिंदी के समान ऊर्दू भी अनेक परसर्ग क्रिया एवं विशेषण के साथ जोड़ती है, जैसे कि – **خوشی** (खुशी), **خوشحالی** (खुशहाली), **خشمجاز** (खुशमिजाज), **خوشنما** (खुशनुमा), **ناخوش** (नाखुश)। ये

विभिन्न रूपांतरित शब्द "**خوش**ا (खुश)" स्टेम के द्वारा बनाये गए हैं। अग्रित खंड में हम कुछ शाब्दिक वर्गों/श्रेणी जैसे **اسم**ا (संज्ञा), **فیل**ا (क्रिया), **صفت** (विशेषण) इत्यादि का संक्षिप्त विवरण प्रदर्शित कर रहे हैं।

A. संज्ञा **(اسم)**: ऊर्दू में संज्ञा को तीन तरह से विभाजित कर सकते हैं।

1. केस : नोमिनेटीव् (Nominative)/ओब्लिक (Oblique) /वोकेटीव् (Vocative) (भाववाचक/अव्यक्त/शब्द वाचक)
2. नंबर : एकवचन /बहुवचन
3. लिंग : पुर्लिंग /स्त्रीलिंग

उदाहरण के लिए :–

- एकवचन पुर्लिंग संज्ञा सामान्यत: ا (अलिफ), ہ (हे) एवं ع (ऍन) के द्वारा समाप्त होती है।
- एकवचन स्त्रीलिंग बनाने के लिए अंत में ی (इ) जोड़ देते हैं।
- बहुवचन नोमिनेटीव् एवं एकवचन ओब्लिक बनाने के लिए आखिरी अक्षर को ے (ये) से प्रतिस्थापित किया जाता है।
- बहुवचन ओब्लिक बनाने के लिए, आखिरी अक्षर को **وں** (ओं) मोर्फीम से प्रतिस्थापित किया जाता है।
- बहुवचन वोकेटीव् बनाने के लिए आखिरी अक्षर को **و** (वोव) से प्रतिस्थापित करते हैं।

तालिका 1– : संज्ञा समूह का उदाहरण

	नोमिनेटिव	ओब्लिक	वोकेटिव
एकवचन	ہتھودا (हथोड़ा)	ہتھودے (हथौड़े)	ہتھودے (हथौड़े)
बहुवचन	ہتھاودے (हथौड़े)	ہتھاودوں (हथौड़ों)	ہتھاوڈو (हथौड़ो)

B. क्रिया **(فیل):** ऊर्दू क्रिया दूसरे शाब्दिक वर्गों कि तुलना में बहुत जटिल है। मार्फोलोजी के सन्दर्भ में, ऊर्दू क्रिया के अनेक रूप हैं –

- लिंग : पुर्लिंग, स्त्रीलिंग
- नंबर : एकवचन, बहुवचन
- व्यक्ति : प्रथम, द्वितीय एवं तृतीय

ऊर्दू क्रिया प्रत्यक्ष एवं अप्रत्यक्ष प्रेरणा का व्यवहार प्रस्तुत करती है। आम तौर पर क्रिया किसी "स्टेम" से ही बनती है। उदाहरण के लिए यदि हम क्रिया 'कर' की बात करें तो–

- साधारण रूप : करना
- प्रत्यक्ष साधारण रूप : कराना
- अप्रत्यक्ष साधारण रूप : करवाना

ये तीनो क्रियाएँ, क्रिया 'कर' की शाब्दिक विच्छेद हैं।

C. विशेषण **(صفت):** ऊर्दू विशेषण भी लिंग, नंबर, एवं कारक के आधार पर इन्फ्लेक्टेड (विभक्त) होते हैं। उदाहरण के लिए – **فرٹیلا** (फुर्तीला) एकवचन पुर्लिंग रूप है, जिसे यदि बहुवचन पुर्लिंग में परिवर्तन करना हो तो **فرتیلے**(फुर्तीले) में परिवर्तित किया जाएगा। यदि अब हम स्त्रीलिंग में परिवर्तित करना चाहे तो हमें **فرتیلی** (फुर्तीली) लिखना होगा।

D. क्रिया–विशेषण **(فیل-صفت)** : क्रिया–विशेषण, विशेषण की तरह ही परिवर्तिनीय एवं अपरिवर्तनीय रूप में होती है। यह संज्ञा एवं क्रिया के रूप परिवर्तन पर आधारित है। हम क्रिया–विशेषण को कुछ निम्नलिखित वर्गों में विभाजित कर सकते हैं।

- समय की क्रिया–विशेषण : **روجانا**ا (रोजाना) / **اکثر** (अक्सर)
- स्थान की क्रिया–विशेषण : **وہاں** (यहाँ) / **ایہاں**(वहाँ)
- व्यवहार की क्रिया–विशेषण : **یکایک** (यकायक)
- कोटि की क्रिया–विशेषण : **چھوٹا**ا (छोटा) / **لمبا**ا (लम्बा)

4. प्रस्तावित प्रणाली

इस प्रस्तावित प्रणाली में हमने नियमबद्ध ऊर्दू स्टेमर का निर्माण किया है। इस नियमबद्ध स्टेमर के द्वारा, इन्फ्लेक्शनल एवं डेरिवेशनल दो तरह के मोर्फीम, "स्टेम" के तौर पर प्राप्त होते हैं। कुछ इन्फ्लेक्शनल एवं डेरिवेशनल मोर्फोलोजी को प्रदर्शित करने वाले शब्द नीचे की तालिका 2 एवम् 3 में दर्शाये गए हैं जो कि हमारे नियमबद्ध स्टेमर के द्व ारा प्राप्त हुए हैं।

तालिका 2 : इन्फ्लेक्शनल मोर्फीम /स्टेम

शब्द	अफिक्स	स्टेम
برسات (बरसात) – संज्ञा	ات (आत)	برس (बरस) – संज्ञा
حمایتی (हिमायती) – संज्ञा	ی (ई)	حمایت (हिमायत) – संज्ञा
حلفنامہ (हलफनाम:) – संज्ञा	نامہ (नाम:)	حلف (हलफ) – संज्ञा

तालिका 3 : डेरिवेशनल मोर्फीम /स्टेम

शब्द	अफिक्स	स्टेम
آبادی (आबादी) – संज्ञा	ی (ई)	آباد (आबाद) – विशेषण
بےحیا ((बेहया) – विशेषण	بےحیا (बे)	حیا (हया) – संज्ञा
خاردار (खारदार) – विशेषण	دار (दार)	خار (खार) – संज्ञा

प्रस्तवित प्रणाली को विकसित करने के लिये हम निम्नलिखित चरणों का पालन करेंगें, जो कि इस तरह से हैं –

4.1 डेटा संग्रह :

ऊर्दू डेटा संग्रह के लिये हमने पर्यटन एवम् स्वास्थ्य विभाग के डेटा को संग्रहित किया है, जिसे निम्नलिखित तालिका 4 में प्रदर्शित किया गया है –

तालिका 4: विस्तारित डेटा संग्रह

डेटासेट	वाक्यों की संख्या	शब्दों की संख्या	3 या 3 से बड़ी लंबाई वाले शब्द
पर्यटन	25000	441617	338745
स्वास्थ्य	25000	441197	339571
कुल	50000	882814	678316

4.2 अफिक्स उत्पन्न :

स्टेमर के विकास के लिए हमने बहुत सारे शब्दों का निरीक्षण किया। इस निरीक्षण से हमें ज्ञात हुआ कि किसी भी शब्द से कुछ प्रेफिक्स एवं सफिक्स जोड़ने एवं हटाने से उनके शब्द विच्छेद में किस किस तरह के परिवर्तन आते हैं। यहाँ हमने 678316 शब्दों के निरीक्षण के पश्चात बिना दोहराये हुए 146 अफिक्स की सूची नियमों के तौर पर तैयार की है, जिनमें से 129 सफिक्स एवं 17 प्रेफिक्स प्राप्त हुए हैं।

ये अफिक्स किसी भी शब्द से स्टेम को पृथक करने के लिए इस्तेमाल किये गए हैं। नीचे के चित्र 1 में कुछ अफिक्सों को दर्शाया गया है।

ے	و	ی	یں	ائیں
ا	ات	ین	نے	تے
یوں	تا	یاں	وں	ائی
گے	ناک	نو	بد	ں

चित्र 1 : नमूना आधारित अफिक्स सूची

उपरोक्त सूची में कई अफिक्स नहीं भी शामिल किया है क्यूंकि कुछ ऐसे भी अफिक्स हैं जिनको

शामिल करने की वजह से ओवरस्टेमिंग की समस्या का सामना करना पड सकता है। इस नियमबद्ध प्रणाली में हम दो तरीके से अफिक्स को स्टेम शब्द से हटा सकते हैं।

4.2.1 प्रणाली 1 : सबसे बड़े प्रत्यय के आधार पर प्रत्यय निष्कासन (Largest suffix based affix removal algorithm)

इस प्रणाली में हम अफिक्स को उसकी लम्बाई के आधार पर घटते क्रम में रखते हैं। यह प्रणाली, स्ट्रिपिंग पद्धति पर आधारित है जो कि अफिक्स को उसके स्टेम शब्द से हटाती जाती है। बड़ी लम्बाई वाले अफिक्स पहले हटेंगे फिर यदि जरूरत होगी तो उससे छोटी लम्बाई वाले अफिक्स हटेंगे। कुछ शब्दों में से हमें उचित स्टेम नहीं भी प्राप्त होता है। उदाहरण के लिए : '**حیات**' (हयात) शब्द से कुछ भी नहीं हटना चाहिए किन्तु जब इस शब्द को अपने स्टेमर से प्रोसेज कराएँगे तो हमें 'ہے' (ह्य) स्टेम एवं सफिक्स 'ات' (आत) अलग हो जाएगा। यह स्टेम कोई अर्थ नहीं रखता है। ऐसे अपवाद शब्दों के लिए हमने एक डेटाबेस तैयार किया है एवम् सभी अप्वाद शब्दों को इसी डेटाबेस में ही समाहित किया हुआ है।

4.2.2 प्रणाली 2 : उच्चतम आवृत्ति के आधार पर प्रत्यय निष्कासन (Highest frequency based affix removal algorithm)

इस प्रणाली में हमने अपने शब्दों से उत्पन्न अफिक्सों की उनके घटते हुए आवृत्ति के आधार पर एक सूची बनाई। इस सूची में उच्चतम आवृत्ति वाले अफिक्सों को सबसे ऊपर और कम आवृत्ति वालों को सूची में सबसे नीचे रखा है, जिसके अंतर्गत उच्चतम आवृत्ति वाले अफिक्स सबसे पहले शब्दों से हटेंगे और फिर उसका स्टेम उत्पन्न करेंगें।

4.3 डेटाबेस :

यह डेटाबेस हमने ऊर्दू के कुछ ऐसे शब्दों के लिए बनाया है जिनमे से कोई भी अफिक्स हटाने कि आवश्यकता नहीं होती है। अर्थात यदि हम इन शब्दों से कोई भी अफिक्स हटाते है तो उनसे मिले स्टेम का कोई अर्थ नहीं रह जाता है। इस डेटाबेस की सहायता से ओवर–स्टेमिंग की समस्या भी कम हुई है। डेटाबेस में रखे हुए अपवाद शब्दों में से कुछ नीचे के चित्र 2 में दर्शाये गए हैं।

لڑکا(लड़का) ,	لڑکی(लड़की) ,	حوالات(हवालात) ,
خرافات(खुराफात),	لڑکے(लड़के),	مرتبا(मरतबा) ,
رتنمالا(रत्नमाला) ,	سپنا(सपना) ,	شادی(शादी) ,
سپرا (सपेरा) ,	نخرا(नखरा) ,	دھوکھے(धोखे)

चित्र 2 : अपवाद शब्द

4.4 एल्गोरिथम :

नियमबद्ध ऊर्दू स्टेमर के लिए निम्नलिखित एल्गोरिथम को प्रस्तावित किया गया है ।

स्टेप–1 : इनपुट ग्रहण कराएँ।

स्टेप–2 : इनपुट चेक करें। इनपुट कोई वाक्य या एकल शब्द हो सकता है।

स्टेप–3 : यदि इनपुट कोई वाक्य है तो उसे शब्दों में टोकेनाइज (विभाजित) करें।

स्टैप–4 : हर एक शब्द (शब्दों की लंबाई 3 या 3 से बडी होनी चाहिये) के लिए निम्नलिखित स्टेप का अनुसरण करें।

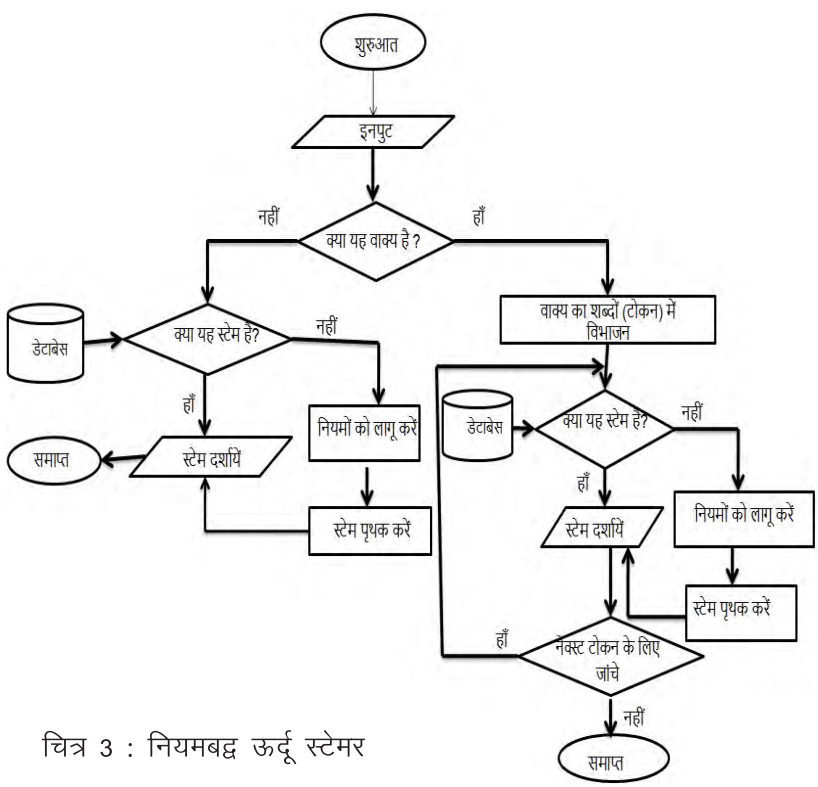
- डेटाबेस से मैच कराये यदि इनपुट डेटाबेस में उपलब्ध है तो स्टेम के समान उसी इनपुट को प्रदर्शित करा देंगे।
- यदि इनपुट डेटाबेस में उपलब्ध नहीं है तो अफिक्स सूची के नियमों को प्रणाली 1 अथवा प्रणाली 2 के आधार पर लागू करा के स्टेम शब्द प्राप्त करेंगे और फिर इस स्टेम को प्रदर्शित कराएँगे।

स्टेप–5 : यदि इनपुट कोई शब्द है तो उपर्युक्त स्टेप– 4 का अनुसरण करें।

अब नीचे की तालिका–5 में इस स्टेमिंग एल्गोरिथम से प्राप्त कुछ परिणाम को प्रदर्शित किया है, आगे हम इस स्टेमिंग एल्गोरिथम को फ्लोचार्ट (चित्र 3) के द्वारा प्रदर्शित कर रहे हैं।

शब्द	स्टेम	प्रेफिक्स	सफिक्स
حضوری (*हुज़ूरी*)	حضور (*हुज़ूर*)		ی (*इ*)
سوالات (*सवालात*)	سوال (*सवाल*)		ات (*आत*)
نظرے (*नज़रे*)	نظر (*नज़र*)		ے (*ए*)
مجلسی (*मजलिसी*)	مجلس (*मजलिस*)		ی (*इ*)
بھولاپن (*भोलापन*)	بھولا (*भोला*)		پن (*पन*)
ایماندار (*ईमानदार*)	امان (*ईमान*)		دار (*दार*)
نوجوان (*नौजवान*)	جوان (*जवान*)	نو (*नौ*)	
لاجواب (*लाजवाब*)	جواب (*जवाब*)	لا (*ला*)	
بدنصیب (*बदनसीब*)	نصیب (*नसीब*)	بد (*बद*)	
بےادب (*बेअदब*)	ادب (*अदब*)	بے (*बे*)	

तालिका – 5:
उर्दू स्टेमर द्वारा प्राप्त परिणाम



चित्र 3 : नियमबद्व ऊर्दू स्टेमर

5. प्रणाली का परिणाम एवम् मूल्यांकन

इस प्रणाली को बनाने के साथ साथ अब हम यह जानना चाहते हैं कि यह प्रणाली कितने प्रतिशत शुद्ध परिणाम देती है अर्थात इस प्रणाली का शुद्ध ता के आधार पर मूल्यांकन करना चाहते हैं। शुद्धता ज्ञात करने के लिए निम्नलिखित सूत्र का इस्तेमाल किया गया है।

शुद्धता (:) = (शुद्ध प्राप्त स्टेम शब्द)/ (कुल इनपुट शब्द) □ 100

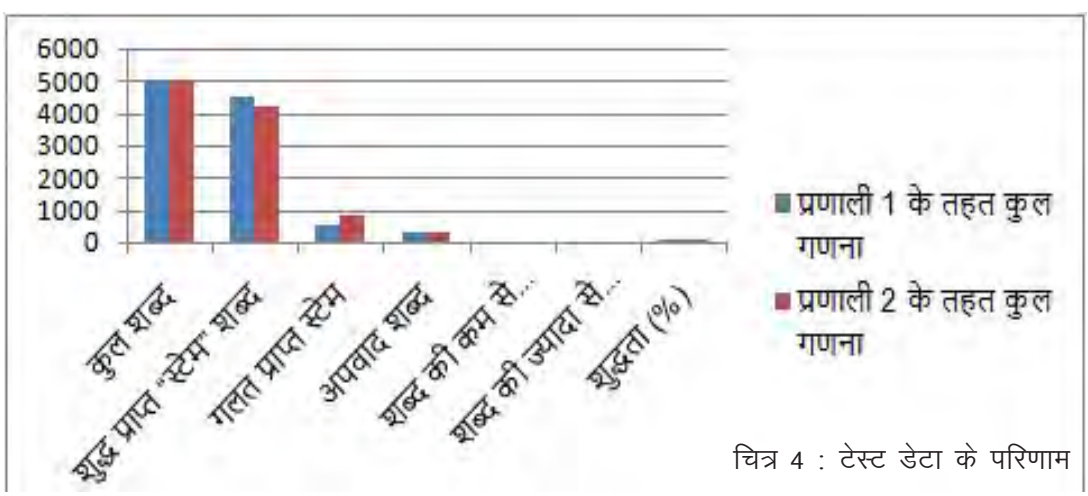
इस शोध पत्र में हमने अपनी प्रणाली की जांच के लिए 5000 शब्दों पर यह एल्गोरिथम लगाई। इन शब्दों को हमने ऊर्दू के लेख से एकाएक चयन किया है। इस जांच वाले डेटा का संक्षिप्त विवरण नीचे की तालिका में दर्शाया गया है।

प्रणाली 1 एवम् प्रणाली 2 के अंतर्गत डेटा का मूल्यांकन करने के लिये हमने इन्हें अपनी एल्गोरिथ्म में लागू किया। इसके पश्चात हमें कुछ इस प्रकार से परिणाम प्राप्त होते हैं जो कि तालिका 6 में दर्शाये गये हैं –

तालिका 6 : प्रणाली 1 एवम प्रणाली 2 पर आधारित जांच डेटा का संक्षिप्त विवरण

जांच डेटा के लक्षण	प्रणाली 1 के अंतर्गत कुल गणना	प्रणाली 2 के अंतर्गत कुल गणना
कुल शब्द	5000	5000
शुद्ध प्राप्त "स्टेम" शब्द	4483	4203
गलत प्राप्त स्टेम	517	797
अपवाद शब्द	312	312
शब्द की कम से कम लम्बाई	3	3
शब्द की ज्यादा से ज्यादा लम्बाई	15	15
शुद्धता (%)	89.6	84.06

उपर्युक्त सूत्र के अनुसार, यदि इस जांच डेटा के द्वारा अपनी प्रणालियों की मैंनुअली (manually) शुद्धता मापते हैं तो हमें 89.60 % एवम 84.06 % की शुद्धता प्राप्त होती है। इन परिणामों से हम यह अनुमान लगा सकते हैं कि प्रणाली 1 जो कि सबसे बड़े प्रत्यय के आधार पर प्रत्यय निष्कासन करती है वो ज्यादा अच्छा एवम् शुद्ध परिणाम प्रदान करती है। नीचे दिया गया चार्ट (चित्र 4) भी उपर्युक्त टेस्ट डेटा के परिणामों को प्रदर्शित करता है।



चित्र 4 : टेस्ट डेटा के परिणाम

6. निष्कर्ष

इस शोध पत्र में हमने नियमबद्ध ऊर्दू स्टेमर के विकास को दर्शाया है और साथ ही कुछ अपवाद शब्दों के लिए डेटाबेस तैयार किया है, जिससे ओवरस्टेमिंग की समस्या कम हुई है। यह नियमबद्ध स्टेमर सिर्फ ऊर्दू भाषा के लिए ही बनाया गया है। इसके द्वारा हम किसी भी शब्द के स्टेम को पृथक कर सकते हैं। इस स्टेमर की शुद्धता प्रणाली 1 के अंतर्गत 89.60 % और प्रणाली 2 के अंतर्गत 84.06 % मापी गई है। भविष्य में हम इस पर और कार्य करके इसकी अफिक्स सूची को और बढ़ाएँगे। इस प्रक्रिया को हम किसी दूसरी भाषा के लिये भी इस्तेमाल कर सकते हैं। किसी अन्य भाषा के लिये हमें अनिवार्य रूप से उनकी अफिक्स सूची और उसी भाषा के अपवादों की सूची तैयार करनी होगी।

Table of English Terminology which is used in paper:

English	Hindi
Algorithm	कलन विधि
Analysis	विश्लेषण
Derivational Morphology	व्युत्पन्न आकृति विज्ञान
Flowchart	प्रवाह संचित्र
Inflectional Morphology	अनिमेश आकृति विज्ञान
Longest Suffix Stripping	सबसे लंबा प्रत्यय अलग करना
Prefix	उपसर्ग
Process	प्रक्रिया
Removal	निष्कासन
Spell Checker	वर्तनी परीक्षक
Suffix	प्रत्यय

सन्दर्भ

[1] Julie Beth Lovins, Development of Stemming Algorithm, Mechanical Translation and Computational Linguistics, *Vol. 11, No. 1, pp 22-31, 1968*

[2] M.F. Porter, An algorithm for suffix stripping, *Program, 14 (3) pp. 130-137. 1980.*

[3] S. Khoja and R. Garside. Stemming Arabic Text. Lancaster, *UK, Computing Department, Lancaster University. 1999.*

[4] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra and K. Datta. "YASS: Yet another suffix stripper", ACM Transactions on Information Systems, Vol. 25, No. 4, pp. 18-38. 2007.

[5] Eiman Tamah Al-Shammari, Jessica Lin, Towards an Error-Free Arabic Stemming, iNEWS'08, *Napa Valley, California, USA. 2008.*

[6] Q. Akram, A. Naseer and S. Hussain, Assas-Band, an Affix- Exception-List Based Urdu Stemmer, In Proceedings of *the 7th Workshop on Asian Language Resources, pp. 40– 47, Suntec, Singapore. 2009.*

[7] Sajjad Ahmad Khan, Waqas Anwar, Usama Ijaz Bajwa, Challenges in Developing a Rule based Urdu Stemmer. In Proceedings of *the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011,, pages 46–51,Chiang Mai, Thailand. 2011.*

[8] Upendra Mishra and Chandra Prakash, MAULIK: An Effective Stemmer for Hindi Language. *International Journal on Computer Science and Engineering (IJCSE), pp. 711-717. 2012.*

[9] Mohd. Shahid Husain. An Unsupervised Approach To Develop Stemmer. *International Journal on Natural Language Computing (IJNLC) Vol. 1, No.2 2012.*

[10] J.Ameta, N.Joshi, I.Mathur. Improving the quality of Gujarati Hindi Machine Translation Through Part-of-Speech Tagging and Stemmer-Assisted Transliteration. In proceeding of *International Journal on Natural Language Computing, Vol 3 (2), pp 49-54,2013.*

[11] S.Paul, M.Tandon, N.Joshi, I.Mathur. Design of a Rule Based Hindi Lemmatizer. In proceeding of the *Third International workshop on Artificial Intelligence, Soft Computing And Application, Chennai, India, pp 67-74, 2013.*

[12] Ali M., Khalid S., and Saleemi M., "A Novel Stemming Approach for Urdu language," Journal of Applied Environmental and Biological Sciences, vol. 4, no. 7S, pp. 436-443, 2014.

[13] Jabbar Abdul, Sajid Iqbal, and Muhammad Usman Ghani Khan. "Analysis and development of resources for Urdu text stemming." Language and Technology 1. 2016.

[14] Mubashir Ali, Shehzad Khalid, and Muhammad Saleemi. Comprehensive Stemmer for Morphologically Rich Urdu Language. The International Arab Journal of Information Technology, Vol. 16, No. 1, pp-139-147, January 2019.

प्रेरक प्रसंग - दीप से दीप जलाओ

ईश्वर चन्द्र विद्यासागर (1820-1891) उन्नीसवीं शताब्दी के बंगाल के प्रसिद्ध दार्शनिक, शिक्षाविद, समाज सुधारक, लेखक, अनुवादक, मुद्रक, प्रकाशक, उद्यमी और परोपकारी व्यक्ति थे। वे बंगाल के पुनर्जागरण के स्तम्भों में से एक थे। उनके बचपन का नाम **ईश्वर चन्द्र बन्दोपाध्याय** था। संस्कृत भाषा और दर्शन में अगाध पाण्डित्य के कारण विद्यार्थी जीवन में ही संस्कृत कॉलेज ने उन्हें 'विद्यासागर' की उपाधि प्रदान की थी। ईश्वर चन्द्र विद्यासागर कलकत्ता में अध्यापन कार्य करते थे। वेतन का उतना ही अंश घर परिवार के लिए खर्च करते जितने में कि औसत नागरिक स्तर का गुजारा चल जाता। शेष भाग वे दूसरे जरूरतमंदों की, विशेषतया छात्रों की सहायता में खर्च कर देते थे। आजीवन उनका यही व्रत रहा।

एक दिन वे बाजार में चले जा रहे थे। एक हताश युवक ने भिखारी की तरह उनसे एक पैसा माँगा। विद्यासागर दानी तो थे पर सत्पात्र की परीक्षा किये बिना किसी की ठगी में न आते। युवक से जवानी में हट्टे कट्टे होते हुए भी भीख माँगने का कारण पूछा। सारी स्थिति जानने पर माँगने का औचित्य लगा। सो एक पैसा तो दे दिया पर उसे रोककर उससे पूछा कि यदि अधिक मिल जाय तो क्या करोगे? युवक ने कहा कि यदि एक रुपया मिले तो उसका सौदा लेकर गलियों में फेरी लगाने लगूंगा और अपने परिवार का पोषण करने में स्वावलम्बी हो जाऊंगा।

विद्यासागर ने एक रुपया उसे और दे दिया। उसे लेकर उसने छोटा व्यापार आरंभ कर दिया। काम दिनों-दिन बढ़ने लगा। कुछ दिन में वह बड़ा व्यापारी बन गया।

एक दिन विद्यासागर उस रास्ते से निकल रहे थे कि व्यापारी दुकान से उतरा, उनके चरण स्पर्श किए, और उन्हें दुकान दिखाने ले गया, और कहा - यह आपकी दी गयी एक रुपये की पूंजी का चमत्कार है। विद्यासागर प्रसन्न हुए और कहा, जिस प्रकार तुमने सहायता प्राप्त करके उन्नति की उसी प्रकार का लाभ अन्य जरूरतमंदों को भी देते रहना। व्यापारी ने वैसा ही करते रहने का वचन दिया।

इस दृष्टांत का संदेश यह है कि उत्साही युवा उद्यमियों को नवाचारमय उद्योग सृजन में प्रोत्साहन पूँजी की व्यवस्था से आत्मनिर्भरता की क्रान्ति संभव है।